

Geant4 visualization with Root-TGeo

Abhisek Sen
Iowa State University

Overview

- ❖ ROOT TGeo is a multi-purpose geometry engine
 - Several useful performance
 - Great visualization
 - Overlap checks
 - Alignment tools
 - Root persistency

- ❖ Future possibility of saving our geometry in TGeo root file (no database).
 - Geometry can be used outside of simulation framework.
 - One can independently run either Geant3/Geant4 or external analysis code.

Visualization

- ❖ vis.mac gives a very poor visualization
 - /vis/ogl/printEPS produces an eps file of ~60MB
 - In RCF it gets distorted
- ❖ Converted our geometry to TGeo file using GDML

← → ↺ gdml.web.cern.ch/GDML/ ☆ ⌵ ☰

Geometry Description Markup Language (GDML)

[Simulation Framework Subproject](#) Last modified 10/09/2015 09:47:19

N E W	09/10/2015	GDML_3_1_3 released
	12/06/2015	GDML_3_1_2 released, Updated User's Manual, release 2.5
	18/11/2014	GDML_3_1_1 released, Updated User's Manual, release 2.4
	26/11/2013	GDML_3_1_0 released, Updated User's Manual, release 2.3
	23/11/2011	GDML_3_0_1 released, Updated User's Manual, release 2.2
	15/12/2010	Updated User's Manual, release 2.1
	18/12/2008	GDML_3_0_0 released

General Information

The Geometry Description Markup Language is an application-indepdent geometry description format based on XML. It can be used as the primary geometry implementation language as well as it provides a geometry data exchange format for the existing applications. The workpackage consists of the GDML Schema part, which is a fully self-consistent definition of the GDML syntax. Since release 3_0_0, the GDML I/O part which provides means for writing out and reading in GDML files is integrated in the packages (like Geant4 and Root) providing GDML compliant interfaces. The GDML Schema does not depend in any way on the GDML I/O parts. In particular, new extensions to the GDML I/O packages can be implemented (interfaces to new application, like for instance visualisation, editors, etc), while the GDML Schema definition remains unchanged.

At the present moment, there exist two toolkit bindings for GDML, the Geant4 binding and the Root binding, both integrated within the respective frameworks. Both bindings support the GDML import (reading GDML files) as well as the export (writing out GDML files).

How to run?

❖ Just run the macro

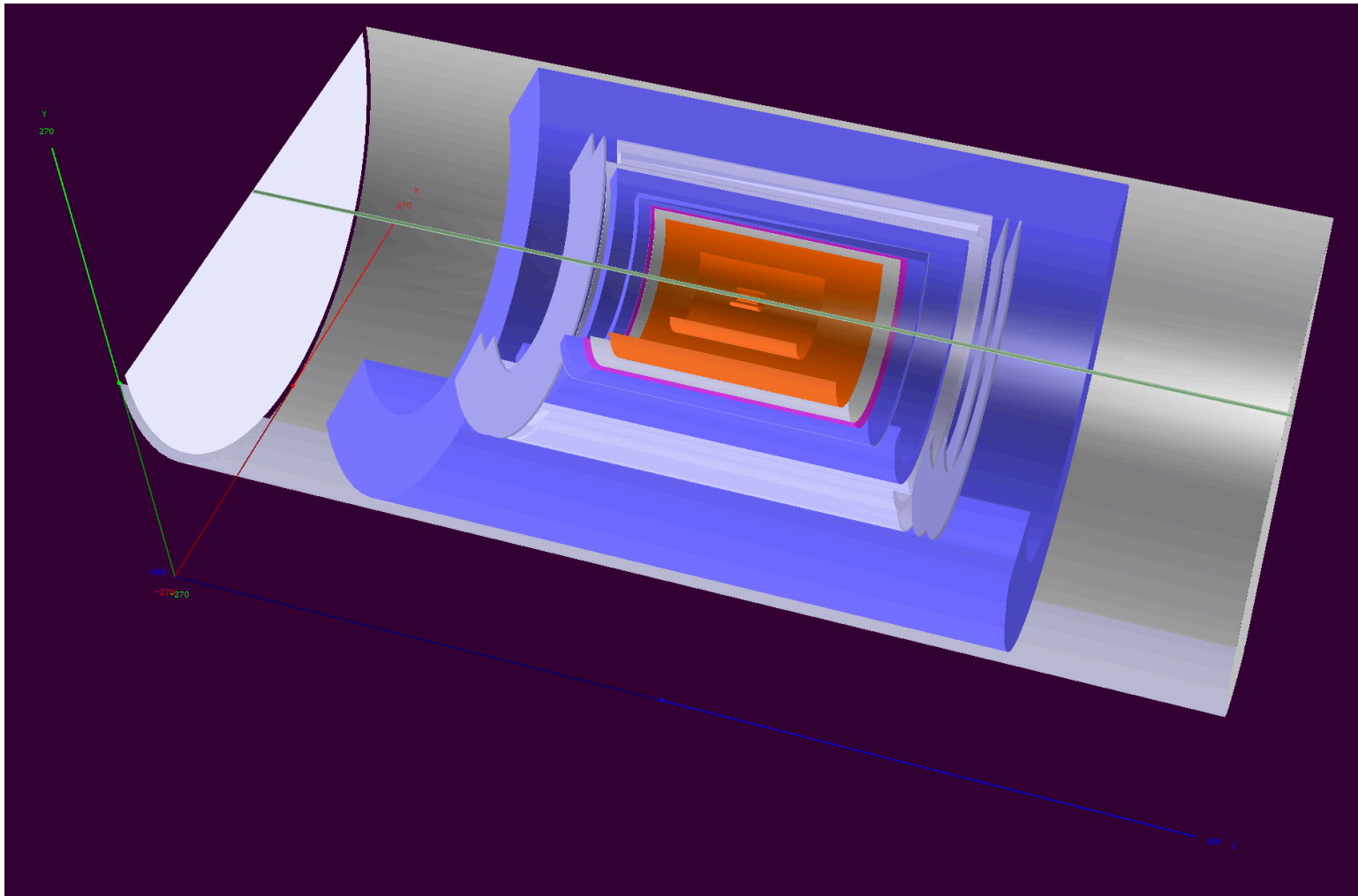
- Geometry file is just 1.5MB, can be read remotely.
- Interactive
- Independent of sphenix environment
- Only requires ROOT
- You can choose a subset of detectors
 - Scenes => Geometry scene => World
- Change Colors
- To see daughter geometries (e.g. Hcal tiles)
 - Hcal_envelop => VizVolumeDaughters
- Overlay with tracks, towers to get great visualization

❖ Run from your laptop (Recommended)

- Assuming you have better graphics card than RCF.

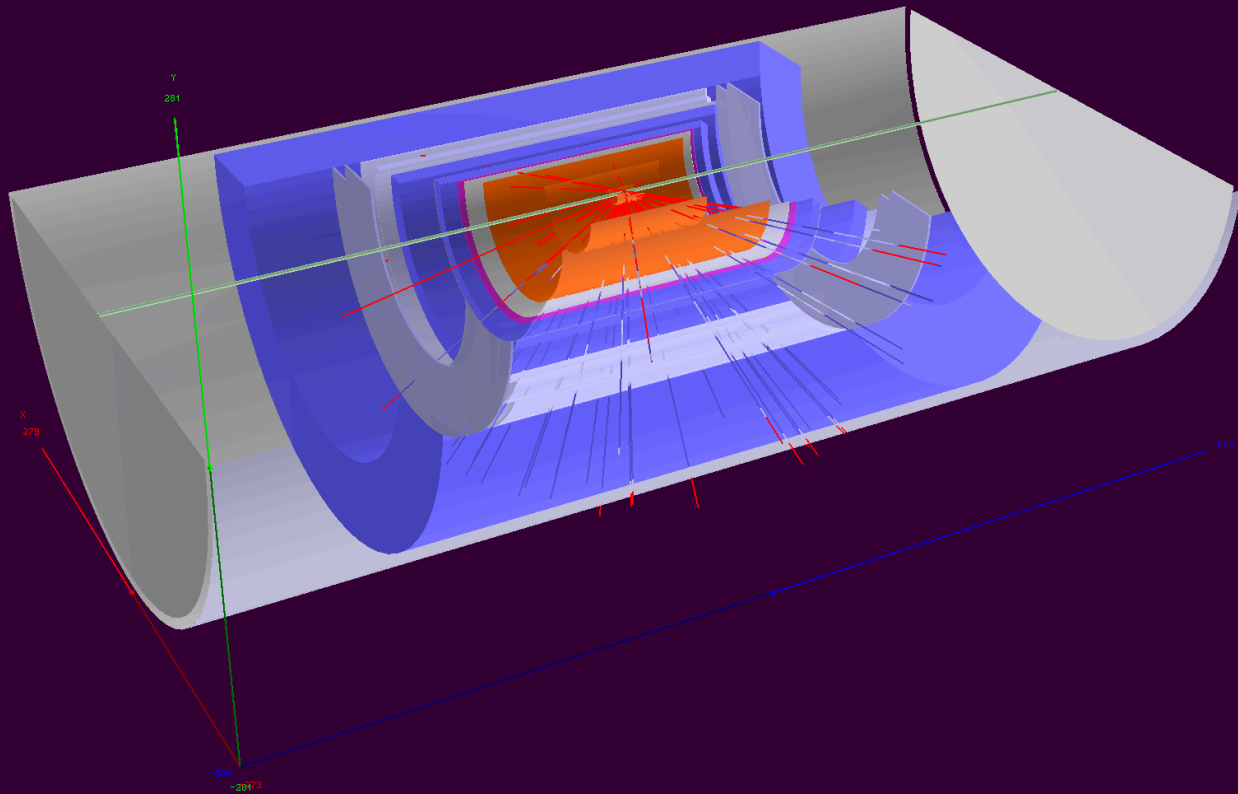
Example 1

`/direct/phenix+sim01/sPHENIX/sunywrk/abhisek/vis/sphenix_geom.C`



Example 2

/direct/phenix+sim01/sPHENIX/sunywrk/abhisek/vis/sphenix_geom_track.C



Example 3

/direct/phenix+sim01/sPHENIX/sunywrk/abhisek/vis/sphenix_geom_jet.C

